UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/632,214	07/31/2003	Gerard Chauvel	TI-35427 (1962-05406)	1113

23494 7590 08/04/2006

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
PETRANEK, JACOB ANDREW	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/632,214	CHAUVEL ET AL.	
	Examiner	Art Unit	
	Jacob Petranek	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 are pending.

2. The office acknowledges the following papers:

Drawings, claims, specification, and arguments filed on 7/19/2006.

Withdrawn objections and rejections

3. The drawing objections are withdrawn due to amendment.

4. The specification objections are withdrawn due to amendment.

5. Claim objections for claims 1-8 and 19-20 are withdrawn due to the specification amendment.

6. The 35 USC § 112 claim rejections for claims 1-8 and 19-20 have been withdrawn due to the specification amendment.

Maintained Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-3, 6, and 19-20 are rejected under 35 U.S.C. §102(b) as being anticipated by Terada et al. (U.S. 6,041,399).

9. As per claim 1:

Terada disclosed a processor executing a plurality of instructions, comprising:

An arithmetic logic unit (Terada: Figure 1 elements 103-104 and 203-204, column 5 lines 35-51); and

A plurality of registers coupled to the ALU, each register programmable to store a register value (Terada: Figure 1 elements 102 and 202, column 5 lines 52-61);

Wherein said processor executes a test and skip instruction that includes an immediate value and a reference to a register, performs a comparison using the immediate value and the register value stored in the referenced register, and selectively skips a subsequent instruction that follows the test and skip instruction based on the comparison (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare greater than instruction in figure 4 compares a register value to an immediate value. The predicate result of this will cause the subtraction instruction to execute or execute and not save the results of the instruction to a register. Thus having the same functionality.).

10. As per claim 2:

Claim 2 essentially recites the same limitations of claim 1. Therefore, claim 1 is rejected for the same reasons as claim 2.

11. As per claim 3:

Terada disclosed the processor of claim 2 wherein the processor skips the subsequent instruction if the immediate value does not match the register value and executes the subsequent if the immediate value does match the register value (Terada: Figure 11, column 5 lines 66-67 continued to column 6 lines 1-24 and column 7 lines 39-48)(Figure 11 shows a compare equal instruction that compares a immediate value

to a register value. The following instruction in figure 11 is skipped if the results aren't equal and is executed if they are equal. Thus having the same functionality.)

12. As per claim 6:

Terada and Blaner disclosed the processor of claim 1 wherein the instruction includes at least one bit that specifies how the comparison is to be performed (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The opcode of the instruction specifies how the instruction will execute. Thus having the same functionality.).

13. As per claim 19:

Terada disclosed a programmable logic device comprising:

Control logic (Terada: Figures 1 and 5, columns 5-6); and

A means for decoding an instruction that includes an immediate value and a reference to a register for performing a comparison using the immediate value and a register value stored in the referenced register, and for causing the processor to execute or skip a subsequent instruction that follows the instruction based on the comparison (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare greater than instruction in figure 4 compares a register value to an immediate value. The predicate result of this will cause the subtraction instruction to execute or execute and not save the results of the instruction to a register. Thus having the same functionality.).

14. As per claim 20:

Terada disclosed the system of claim 19 including means for comparing the immediate value to the register value in the referenced register (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare greater than instruction in figure 4 compares a register value to an immediate value. The predicate result of this will cause the subtraction instruction to execute or execute and not save the results of the instruction to a register. Thus having the same functionality.).

Maintained Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. §103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 4-5 and 7-9 are rejected under 35 U.S.C. §103(a) as being unpatentable over Terada et al. (U.S. 6,041,399), in view of Blaner et al. (U.S. 5,659,722).

17. As per claim 4:

Terada disclosed the processor of claim 1.

Terada failed to teach wherein the comparison is performed by masking the register value in the referenced register with the immediate value and examining one or more bits in the masked version of the referenced register.

However, Blaner disclosed wherein the comparison is performed by masking the register value in the referenced register with the immediate value and examining one or more bits in the masked version of the referenced register (Blaner: Figures 3 and 7,

column 4 lines 53-67 continued to column 5 lines 1-42 and column 6 lines 26-39 and column 8 lines 4-31)(Blaner disclosed a branch instruction that tests the status registers with a immediate mask in the instruction. Figure 3 shows the instruction format, with element 508 being the masked immediate value. Figure 7 shows the process of executing this instruction. Element 330 is the status register that stores the current status bits that are checked by the branch instruction. Column 8 discusses the process of masking the immediate mask in the instruction with the status register values to determine test value. This test value is the basis of if the branch is taken or not.).

Processing elements produce condition signals during execution that indicate conditions relevant to the execution of an instruction. These signals can later be used by conditional branches to determine if certain conditions were met for a branch instruction (Bladen: Column 1 lines 20-29). This type of branch instruction that performs a mask of the status bits with an immediate value to determine if the branch is taken would also be useful in other processors, such as the processor of Terada. The advantage of having conditional branches that branch on the status bits generated from prior instructions would have motivated one of ordinary skill in the art at the time of the invention to implement the test and branch instruction of Bladen into the processor of Terada. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to add the test and branch instruction to Terada for the benefits of branching dependent on different status bits generated from previous instructions.

It would have been obvious to one of ordinary skill in the art that having an instruction with two different modes of operation using a control bit is essentially the

same as having two different instructions using two separate opcodes to control two different modes of operation. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the two instructions and use a single control bit in the opcode to differentiate between the two modes of execution. In addition, according to "In re Larson" (144 USPQ 374 (CCPA 1965)), to make integral doesn't give patentability over prior art.).

18. As per claim 5:

Terada and Blaner disclosed the processor of claim 4 wherein the masking is performed by ANDing the immediate value with the register value (Blaner: Figure 7 elements 352, 362, 372, and 382, column 8 lines 4-31)(These elements are ANDing the value from the status register with the immediate value of the predicate from the branch instruction.).

19. As per claim 7:

Terada disclosed the processor of claim 6 wherein the at least one bit specifies whether the register reference is to a register from a first group of registers or to a register from a second group of registers (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The first group of registers is registers from the register file and the second group of registers is the status register. The compare instruction's opcode specifies that a comparison will be done by a normal register and not a status register. Thus having the same functionality.); and

if a register from the first group of registers is specified by said bit, the comparison is performed by comparing the immediate value to the register value

(Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare instruction in figure 4 is done by comparing the data stored in the register to the immediate value. Thus having the same functionality.); and

Terada failed to teach if a register from the second group of registers is specified by said bit, the comparison is performed by masking the register value with the immediate value and examining one or more bits in the masked version of the referenced register.

However, Bladen disclosed if a register from the second group of registers is specified by said bit, the comparison is performed by masking the register value with the immediate value and examining one or more bits in the masked version of the referenced register (Blaner: Figures 3 and 7, column 4 lines 53-67 continued to column 5 lines 1-42 and column 6 lines 26-39 and column 8 lines 4-31)(The test and branch instruction's opcode states that the comparison is done between the status register and the immediate predicate value within the instruction, which is element 508 in figure 3. Figure 7 shows the process of executing this instruction. Element 330 is the status register that stores the current status bits that are checked by the branch instruction. Column 8 discusses the process of masking the immediate mask in the instruction with the status register values to determine test value. This test value is the basis of if the branch is taken or not.).

Processing elements produce condition signals during execution that indicate conditions relevant to the execution of an instruction. These signals can later be used by conditional branches to determine if certain conditions were met for a branch

instruction (Bladen: Column 1 lines 20-29). This type of branch instruction that performs a mask of the status bits with an immediate value to determine if the branch is taken would also be useful in other processors, such as the processor of Terada. The advantage of having conditional branches that branch on the status bits generated from prior instructions would have motivated one of ordinary skill in the art at the time of the invention to implement the test and branch instruction of Bladen into the processor of Terada. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to add the test and branch instruction to Terada for the benefits of branching dependent on different status bits generated from previous instructions.

It would have been obvious to one of ordinary skill in the art that having an instruction with two different modes of operation using a control bit is essentially the same as having two different instructions using two separate opcodes to control two different modes of operation. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the two instructions and use a single control bit in the opcode to differentiate between the two modes of execution. In addition, according to "In re Larson" (144 USPQ 374 (CCPA 1965)), to make integral doesn't give patentability over prior art.).

20. As per claim 8:

Terada disclosed the processor of claim 6 wherein if the register reference specified by said at least one bit is not the status register, the comparison is performed by comparing the immediate value to the register value in the referenced register (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare

instruction in figure 4 is done by comparing the data stored in the specified register to the immediate value. The register is a normal register within the register file and is not the status register. Thus having the same functionality.)

Terada failed to teach wherein the registers include a status register, and, if the register reference specified by said at least one bit is the status register, the comparison is performed by masking the register value in the status register with the immediate value and examining one or more bits in the masked version of the status register.

However, Bladen disclosed wherein the registers include a status register (Bladen: Figure 5, column 7 lines 1-33)(The multiple predicate register stores predicate value for instructions that don't have an immediate predicate value and also stores status bits for each processing element. Thus having the same functionality.); and

If the register reference specified by said at least one bit is the status register, the comparison is performed by masking the register value in the status register with the immediate value and examining one or more bits in the masked version of the status register (Blaner: Figures 3 and 7, column 4 lines 53-67 continued to column 5 lines 1-42 and column 6 lines 26-39 and column 8 lines 4-31)(The test and branch instruction's opcode states that the comparison is done between the status register and the immediate predicate value within the instruction, which is element 508 in figure 3. Figure 7 shows the process of executing this instruction. Element 330 is the status register that stores the current status bits that are checked by the branch instruction. Column 8 discusses the process of masking the immediate mask in the instruction with the status register values to determine test value. This test value is the basis of if the

branch is taken or not.).

Processing elements produce condition signals during execution that indicate conditions relevant to the execution of an instruction. These signals can later be used by conditional branches to determine if certain conditions were met for a branch instruction (Bladen: Column 1 lines 20-29). This type of branch instruction that performs a mask of the status bits with an immediate value to determine if the branch is taken would also be useful in other processors, such as the processor of Terada. The advantage of having conditional branches that branch on the status bits generated from prior instructions would have motivated one of ordinary skill in the art at the time of the invention to implement the test and branch instruction of Bladen into the processor of Terada. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to add the test and branch instruction to Terada for the benefits of branching dependent on different status bits generated from previous instructions.

It would have been obvious to one of ordinary skill in the art that having an instruction with two different modes of operation using a control bit is essentially the same as having two different instructions using two separate opcodes to control two different modes of operation. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the two instructions and use a single control bit in the opcode to differentiate between the two modes of execution. In addition, according to "In re Larson" (144 USPQ 374 (CCPA 1965)), to make integral doesn't give patentability over prior art.).

21. As per claim 9:

Terada disclosed a method of executing an instruction having a reference to a register, an immediate value, and a control bit that dictates one of at least two tests, the method comprising:

Examining said control bits to determine its state (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The opcode of the instruction specifies how the instruction will execute.);

If said control bits are in a first state, comparing the immediate value to the contents of the register referenced in the instruction and skipping a subsequent instruction based on the outcome of the comparison (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare instruction in figure 4 is done by comparing the data stored in the specified register to the immediate value. The register is a normal register within the register file and is not the status register. Thus having the same functionality.); or

Terada failed to teach if said control bit is in a second state, masking the contents of the register with the immediate value, testing one or more bits in the masked version of the contents of the register, and skipping a subsequent instruction based on the outcome of the testing.

However, Bladen disclosed if said control bits are in a second state, masking the contents of the register with the immediate value, testing one or more bits in the masked version of the contents of the register, and skipping a subsequent instruction based on the outcome of the testing (Blaner: Figures 3 and 7, column 4 lines 53-67 continued to column 5 lines 1-42 and column 6 lines 26-39 and column 8 lines 4-31)(The test and

branch instruction's opcode states that the comparison is done between the status register and the immediate predicate value within the instruction, which is element 508 in figure 3. Figure 7 shows the process of executing this instruction. Element 330 is the status register that stores the current status bits that are checked by the branch instruction. Column 8 discusses the process of masking the immediate mask in the instruction with the status register values to determine test value. This test value is the basis of if the branch is taken or not.).

Processing elements produce condition signals during execution that indicate conditions relevant to the execution of an instruction. These signals can later be used by conditional branches to determine if certain conditions were met for a branch instruction (Bladen: Column 1 lines 20-29). This type of branch instruction that performs a mask of the status bits with an immediate value to determine if the branch is taken would also be useful in other processors, such as the processor of Terada. The advantage of having conditional branches that branch on the status bits generated from prior instructions would have motivated one of ordinary skill in the art at the time of the invention to implement the test and branch instruction of Bladen into the processor of Terada. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to add the test and branch instruction to Terada for the benefits of branching dependent on different status bits generated from previous instructions.

Terada and Bladen failed to teach a control bit that specifies a first or second state.

However, it would have been obvious to one of ordinary skill in the art that having

an instruction with two different modes of operation using a control bit is essentially the same as having two different instructions using two separate opcodes to control two different modes of operation. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the two instructions and use a single control bit in the opcode to differentiate between the two modes of execution. In addition, according to "In re Larson" (144 USPQ 374 (CCPA 1965)), to make integral doesn't give patentability over prior art.).

22. Claim 10 is rejected under 35 U.S.C. §103(a) as being unpatentable over Terada et al. (U.S. 6,041,399), in view of Blaner et al. (U.S. 5,659,722), further in view of Chen et al. (U.S. 5,504,903)

23. As per claim 10:

Terada and Bladen disclosed the method of claim 9.

Terada and Bladen failed to teach wherein skipping the subsequent instruction comprises replacing the subsequent instruction with a no operation instruction.

However, Chen disclosed wherein skipping the subsequent instruction comprises replacing the subsequent instruction with a no operation instruction (Chen: Column 7 lines 59-67 continued to column 8 lines 1-3).

Both the bit test and skip if set/clear instructions are essentially a predicated compare instruction, which will only execute the next instruction if a condition is met. If the condition is met, then the next instruction is not allowed to complete and is essentially the same as a nop instruction. Thus, it would have been obvious to one of

ordinary skill in the art to use the process from Chen of substituting in a nop instruction instead of the instruction from Bladen or Terada that will simply not complete if the condition to not execute is met.

New Claim Rejections - 35 USC § 103

24. The following is a quotation of 35 U.S.C. §103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

25. Claims 11-13, 16, and 18 are rejected under 35 U.S.C. §103(a) as being unpatentable over Terada et al. (U.S. 6,041,399), in view of Feierbach et al. (U.S. 6,088,786).

26. As per claim 11:

Terada disclosed a system, comprising:

A main processor unit (Terada: Figure 16 figure 20, column 10 lines 43-49); and

A co-processor coupled to said main processor unit (Terada: Figure 16 element 22, column 10 lines 43-49);

Wherein, during the register-based instruction mode, the coprocessor executes an instruction that includes an immediate value and a reference to a register accessible to said co-processor, performs a comparison using the immediate value and the register value, and executes or skips a subsequent instruction based on the comparison (Terada: Figure 4, column 5 lines 66-67 continued to column 6 lines 1-24)(The compare

greater than instruction in figure 4 compares a register value to an immediate value. The predicate result of this will cause the subtraction instruction to execute or execute and not save the results of the instruction to a register. Thus having the same functionality. Since the instruction deals with registers, the coprocessor executes the comparison instruction in a register-based mode.).

Terada fails to teach wherein said co-processor selectively operates in a stack-based instruction mode and a register-based instruction mode.

However, Feierbach disclosed wherein said co-processor selectively operates in a stack-based instruction mode and a register-based instruction mode (Feierbach: Figure 2 element 227, column 7 lines 27-37)(Figure 2 shows a processor that is able to selectively execute stack-based instructions and register-based instructions by using a predecoder, element 227, to determine where the current instruction is to go.).

The advantage of stack-based processors is that they are much more compact and efficient than there register-based counterparts. Having both a stack-based and register-based processor is advantageous when a processor also has to occasionally execute high-performance multimedia applications, which are better suited for register-based processors (Feierbach: Column 2 lines 44-67 continued to column 3 lines 1-45). One of ordinary skill in the art would have been motivated by the increased performance in certain applications for stack-based processors to add a stack-based processor to the processor of Terada. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a stack-based processor alongside the register-based processor of Terada for the advantage of increased performance in

certain applications.

27. As per claim 12:

Claim 12 essentially recites the same limitations of claim 2. Therefore, claim 12 is rejected for the same reasons as claim 2.

28. As per claim 13:

Claim 13 essentially recites the same limitations of claim 3. Therefore, claim 13 is rejected for the same reasons as claim 3.

29. As per claim 16:

Claim 16 essentially recites the same limitations of claim 6. Therefore, claim 16 is rejected for the same reasons as claim 6.

30. As per claim 18:

The system of claim 11 further comprising wireless communication circuitry and said system comprises a cell phone (Official notice is taken that the processing system could be part of a cellular telephone.).

31. Claims 14-15 and 17 are rejected under 35 U.S.C. §103(a) as being unpatentable over Terada et al. (U.S. 6,041,399), in view of Blaner et al. (U.S. 5,659,722), further in view of Feierbach et al. (U.S. 6,088,786).

32. As per claim 14:

Claim 14 essentially recites the same limitations of claim 4. Therefore, claim 14 is rejected for the same reasons as claim 4.

33. As per claim 15:

Claim 15 essentially recites the same limitations of claim 5. Therefore, claim 15 is rejected for the same reasons as claim 5.

34. As per claim 17:

Claim 17 essentially recites the same limitations of claim 7. Therefore, claim 17 is rejected for the same reasons as claim 7.

Response to Arguments

35. The arguments presented by Applicant in the response, received on 7/19/2006 are not considered persuasive.

36. Applicant argues that "Terada failed to teach a test and skip instruction as claimed" for claims 1 and 19.

This argument is not found to be persuasive for the following reason. The compare instruction of figure 4 is read as the test and skip instruction. The compare instruction compares an immediate value and a register value. Based on this comparison, the next instruction is selectively skipped by checking the predicate register.

For claim 19, the instruction being executed is defined as the pipeline executing the instruction and committing the instruction to memory. If the instruction isn't committed to memory, then the instruction isn't fully executed. Thus, this interpretation is similar to an instruction being skipped by the specification's definition.

37. Applicant argues that "Terada doesn't disclose whether the instruction actually includes the immediate value" for claim 1.

This argument is not found to be persuasive for the following reason. The comparison instruction in figure 4 has an immediate field, and therefore inherently has an immediate field that is compared to a register.

38. Applicant argues that "A control bit that specifies a first or second state isn't obvious" in claim 9.

This argument is not found to be persuasive for the following reason. Applicant makes many arguments in favor of nonobviousness.

The first is that the applicants instruction can be decoded within one cycle, compared to using two instructions. The instruction with the control bit will only execute one specific test based on the control bit, thus it wouldn't make sense to have the two separate instruction included within the program if only one is going to be used.

The second is storing a instruction with a control bit requires less memory space than storing two instructions. Again, the instruction with the control bit will only execute one specific test based on the control bit, thus it wouldn't make sense to have the two separate instruction included within the program if only one is going to be used.

The third is decoding an instruction requires planning as to what the control bit will mean. Careful planning is done when designing the decode stage because all instructions must be mapped out and checked by comparators for opcode and/or various control bits embedded within instructions so that the processor can properly execute the instruction. This fact is obvious to one of ordinary skill in the art at the time of the invention.

39. Applicant argues that "Blaner's masking is not based on an immediate value provided with an instruction" for claim 9.

This argument is not found to be persuasive for the following reason. Figure 7 shows the process of the masking, testing, and skipping a subsequent instruction when in combination with Terada. Blaner's branch instruction shows the masking of an immediate value, being PE1-4 (Figure 3 shows that any of the PE1-4, being element 508, is an immediate value), is compared or masked to the PR4-1 values in the test logic. One of ordinary skill in the art would realize that the Test Logic, element 302, would use an AND operator to compare the PE1-4 and PR4-1 elements. This is because the PE1-4 is used to determine if a particular condition if met, as specified by table 1 in the specification. This is compared to see if the condition is matched in the status register that stores the PR4-1 values. The AND operator would be a likely source to determine if they match, which results in masking the 4 bit values of each down to a single bit value that will be later tested by the OR operators to see if a branch occurs. If a branch does occur, then the next instruction can be skipped according to Terada's predicated execution.

40. Applicant argues that "Terada and Blaner failed to teach wherein said coprocessor selectively operates in a stack-based instruction mode and a register-based instruction mode."

This argument is found to be persuasive for the following reason. However, due to the amendment, a new grounds of rejection has been given for the claims.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Patel et al. (U.S. 6,826,749), taught a processor that executes stack-based and register-based instructions by translating stack-based instructions into register-based instructions.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jacob Petranek
Examiner
Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100